

# Open MPI调研报告

摘要：Open MPI作为MPI(Message Passing Interface)标准的实现之一，实现了不同进程间的点对点或广播通信。它具有可拓展性，支持[大量](#)节点间的通信。相较另外两种通用实现（MPICH2, Intel MPI），它还具有多平台支持、免费的特点。

作者：王琦

日期：2015-12-25

根据[boost官方](#)，boost.MPI只在三种MPI实现上进行过测试：

- Open MPI
- MPICH2
  - 根据调查，MPICH在1.4版本后不再支持windows
    - [MPICH官方下载](#)中只给出了基于1.0.3版本的MS版本
    - [stackoverflow上的回答](#)说MPICH对windows的版本只支持到1.4
- Intel MPI
  - [付费](#)。499\$一套。

所以我调研了OpenMPI.

“

## 如何安装**Open MPI**?

[How do I build Open MPI?](#)

源码：<http://mirror.its.dal.ca/openmpi/software/ompi/v1.10/>

- 对于windows平台，下载cygwin版本，**可能能用**。我得周一来实验室看看。

在官网下载源码，解压并编译：

```
$ gunzip -c openmpi-1.10.1.tar.gz | tar xf -
$ cd openmpi-1.10.1
$ ./configure --prefix=/usr/local
<...lots of output...>
$ make all install
```

“

## 如何指定运行**MPI**工作的主机?

## [How do I specify the hosts on which my MPI job runs?](#)

有三种通用机制用于制定MPI任务运行的主机：

- --hostfile选项
- --host选项
- 在调度环境（e.g. SLURM, Torque, 或LFS job）下，Open MPI会自动从调度器中获得主机列表。

“

### **在mpirun命令中如何使用--hostfile选项？**

## [How do I use the --hostfile option to mpirun?](#)

--hostfile选项指定mpirun所用的hostfile, hostfile用于指定哪些主机发起进程。hostfile内容可以是：

```
# The following node is a single processor machine:
foo.example.com

# The following node is a dual-processor machine:
bar.example.com slots=2

# The following node is a quad-processor machine, and we absolutely
# want to disallow over-subscribing it:
yow.example.com slots=4 max-slots=4
```

slots:

max-slots:

Hostfiles有两种工作方式：

- Exclusionary: 如果主机列表（hostlist）已由**调度器**给定，那么hostfile里的节点必须是它的子集，否则mpirun会被中止。
- Inclusionary: 如果主机列表（hostlist）并未被给定，在hostfile中则可以包含任意主机。

“

### **如何在不同节点上调度openmpi进程？(--host选项)**

## [How do I control how my processes are scheduled across nodes?](#)

--host选项和--hostfile选项类同。

“

**在不使用`hostfile`的情况下，`slot`数是如何被计算的？**

[\*I'm not using a hostfile. How are slots calculated?\*](#)

如果你在使用被支持的资源管理器，Open MPI会直接从实体中获得slot信息。

```
$ mpirun --host host node0, node0, node0, node0
```

这告诉Open MPI, node0有4个slot, 这和下述命令不同：

```
$ mpirun -np 4 --host node0 a.out
```

这告诉Open MPI, node0有1个slot, 但你在上面运行了4个进程。这显式通知了Open MPI你在超额订购（oversubscribing）该节点。

“

**我可以在单处理器机器上运行多个并行线程吗？**

[\*Can I run multiple parallel processes on a uniprocessor machine?\*](#)

可以。

这是在oversubscribing节点。但一定要确保Open MPI知道你在oversubscribing节点，否则这将造成严重的性能退化。

“

**我可以超额订购节点吗？（运行比处理器数目更多的进程）**

[\*Can I oversubscribe nodes \(run more processes than processors\)?\*](#)

可以。

但一定要确保Open MPI知道你在oversubscribing节点，否则这将造成严重的性能退化。

指导性原则是：**永远不要指定多于核数的slot数**。比如，你想在单核上运行4个进程，那么应指出你只有一个slot但想运行4个进程，如：

```
$ cat my-hostfile
localhost
$ mpirun -np 4 --hostfile my-hostfile a.out
```

**不要**在你的hostfile中出现"slots = 4"（应为你只有一个核）

Open MPI有两种运行它的消息传递演进引擎（message passing progression engine）的模式：

- Degraded: 当Open MPI认为它在oversubscribing模式下运行，进程会主动向其它peer交出CPU
- Aggressive: 当Open MPI认为它在exact- or under-subscribed 模式下运行时，MPI进程会自动的在aggressive模式下运行，**永远不愿意**交出CPU.

考虑上述在单核机器上制定4个slot的情况：

```
$ cat my-hostfile
localhost slots=4
$ mpirun -np 4 --hostfile my-hostfile a.out
```

这会导致所有4个MPI进程在aggressive模式下运行，因为Open MPI认为它拥有4个处理器。这将导致严重的性能损失。